

CLAIMS:

1. A method for recovering a high level structure of a target program,
comprising the acts of:
 - 5 a) generating text detection data for the target program;
 - b) generating a genre/sub-genre feature vector for the target program
using the text detection data generated at act (a);
 - c) creating a plurality of high order graphical models;
 - 10 d) identifying a subset of said high order graphical models using said
target program cluster distribution data; and
 - d) identifying a single high order graphical model from said subset of
models using said target program text detection data,
wherein said single high order graphical model corresponds to the high
level structure of the target program.
- 15 2. The method of Claim 1, further comprising the act of creating a
program summary using said single high order graphical model with said text
detection data.
- 20 3. The method of Claim 2, wherein said act of creating said program
summary further comprises the acts of:
 - determining one or more events of importance to a viewer;
 - searching said text detection data for said important events;
 - extracting said important events from said text detection data; and
 - 25 including said extracted events in said program summary.
4. The method of Claim 1, further comprising the act of creating a
program summary, comprising the acts of:
 - 30 searching for a program event;
 - ranking the program events identified at said searching act
based on a pre-determined ranking;
 - selecting certain of said identified program events based on
said ranking.

5. The method of Claim 4, wherein the act of searching for a program event comprises the acts of:
- determining a sequence of text events that collectively define a program event;
 - searching said text detection data for said sequence of text events;
 - upon identifying said sequence of text events in said text detection data, comparing said sequence of text events with corresponding nodes in said high order graphical model; and
 - determining if the time sequence of occurrence of said sequence of text events conforms with time constraints associated with said corresponding nodes in said high order graphical model.
6. The method of Claim 1, further comprising the act of searching for information in said target program comprising: text types, similarities with programs other than said target program, text patterns, program events and patterns of program events.
7. The method of Claim 6, wherein said information to be searched in said target program uses information provided by said text detection data and said single high order graphical model.
8. The method of Claim 1, wherein said graphical model is one of a Petri net model, a Hidden Markov Model and a combination of said Petri net model and said Hidden Markov Model.
9. The method of Claim 1, wherein the target program is one of a television and video program.
10. The method of Claim 1, wherein the act of generating text detection data for the target program further comprises the acts of:
- i) detecting the presence of text in the target program;
 - ii) identifying and extracting text features of the detected text; and
 - iii) forming text feature vectors from the identified and extracted

features.

11. The method of Claim 10, wherein the act of detecting the presence of text
5 in the target program is performed in accordance with the MPEG-7 standard.

12. The method of Claim 10, wherein said identified and extracted text features comprise text position, text height, text font type, and text color.

10 13. The method of Claim 10, wherein the act of detecting the presence of text
in the target program further comprises the act of detecting the presence of text in particular video frames of the target program.

15 14. The method of Claim 10, wherein the act of generating said genre/sub-genre feature vector for the target program further comprises the acts of:
comparing the text feature vectors for the target program generated at
act
(iii) with a plurality of pre-determined genre/sub-genre feature vectors for various
20 genre/sub-genre types; and
associating the text feature vectors for the target program with the
genre/sub-genre feature vectors having the highest degree of similarity;
defining the collection of genre/sub-genre feature vectors identified at
the
25 associating step as the genre/sub-genre feature vector for the target program. .

15. The method of Claim 1, wherein said plurality of high order graphical models graphically model particular program genre/sub-genre types at a program level.
30

16. The method of Claim 12, wherein a transition element of said high order graphical model may be comprised of a low order graphical model, said low order model including program text and timing information.

17. The method of Claim 16, wherein said low order graphical model is modeled as a Petri net.

18. The method of Claim 17, wherein said transition element may be assigned a priority rank order relative to other transition elements of said high order model.

19. The method of Claim 1, wherein the act of generating genre feature vector clusters data for the target program is performed in accordance with an unsupervised clustering algorithm.

20. The method of Claim 19, wherein the unsupervised clustering algorithm is based on a distance metric comparing corresponding text features.

21. The method of Claim 20, wherein the distance metric is computed as:

$$\begin{aligned} \text{Dist}(F_{V1}, F_{V2}) = & w1 * (|F_{V1\text{row}} - F_{V2\text{row}}| + |F_{V1\text{col}} - F_{V2\text{col}}|) + \\ & w2 * (|F_{V1h} - F_{V2h}| + \\ & w3 * (|F_{V1f} - F_{V2f}| + |F_{V1g} - F_{V2g}| + |F_{V1b} - F_{V2b}|) + \\ & w4 * (\text{FontDist}(f1, f2)) \end{aligned}$$

where:

$F_{V1\text{row}}, F_{V2\text{row}} = 1^{\text{st}}$ and 2^{nd} feature vector row positions;

$F_{V1\text{col}}, F_{V2\text{col}} = 1^{\text{st}}$ and 2^{nd} feature vector column positions;

$F_{V1h}, F_{V2h} = 1^{\text{st}}$ and 2^{nd} feature vector heights;

$F_{V1r}, F_{V1g}, F_{V1b} = 1^{\text{st}}$ feature vector color (r,g,b);

$F_{V2r}, F_{V2g}, F_{V2b} = 2^{\text{nd}}$ feature vector color (r,g,b);

$f1 =$ font type of first feature vector;

$f2 =$ font type of second feature vector; and

$\text{FontDist}(a,b) =$ A precomputed distance between multiple font types.

22. A system for recovering the high-level structure of a target program,

- said system comprising: a memory for storing computer readable code, a database for storing a plurality of higher-order Petri nets and a processor operatively coupled to said memory, said processor configured to generate text detection data for the target program; generate a genre/sub-genre feature vector for the target program using the
- 5 text detection data; create a plurality of high order graphical models; identify a subset of said high order graphical models using said target program cluster distribution data; and identify a single high order graphical model from said subset of models using said target program text detection data, wherein said single high order graphical model corresponds to the high
- 10 level structure of the target program.